Machine learning of a Higgs decay classifier via quantum annealing

...

Presenter: Joshua Job[1]

Reference: “Solving a Higgs optimization problem with quantum annealing for machine learning”, forthcoming, *Nature*

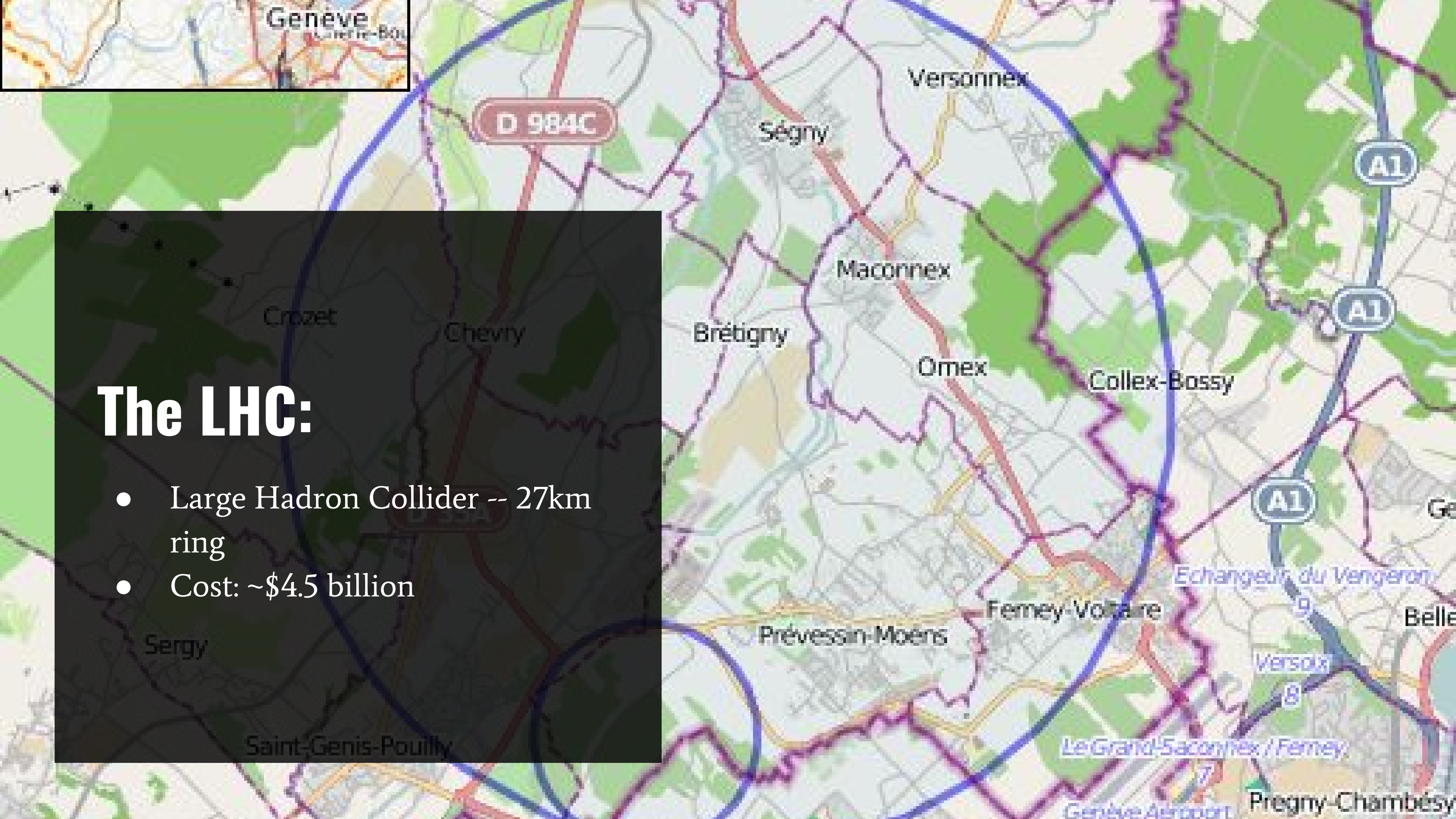
Collaborators: Alex Mott[2], Jean-Roch Vlimant[2], Daniel Lidar[3], Maria Spiropulu[2]

Associations: 1. Department of Physics, Center for Quantum Information Science & Technology, University of Southern California 2. Department of Physics, California Institute of Technology 3. Departments of Electrical Engineering, Chemistry, and Physics, Center for Quantum Information Science & Technology, University of Southern California

Outline

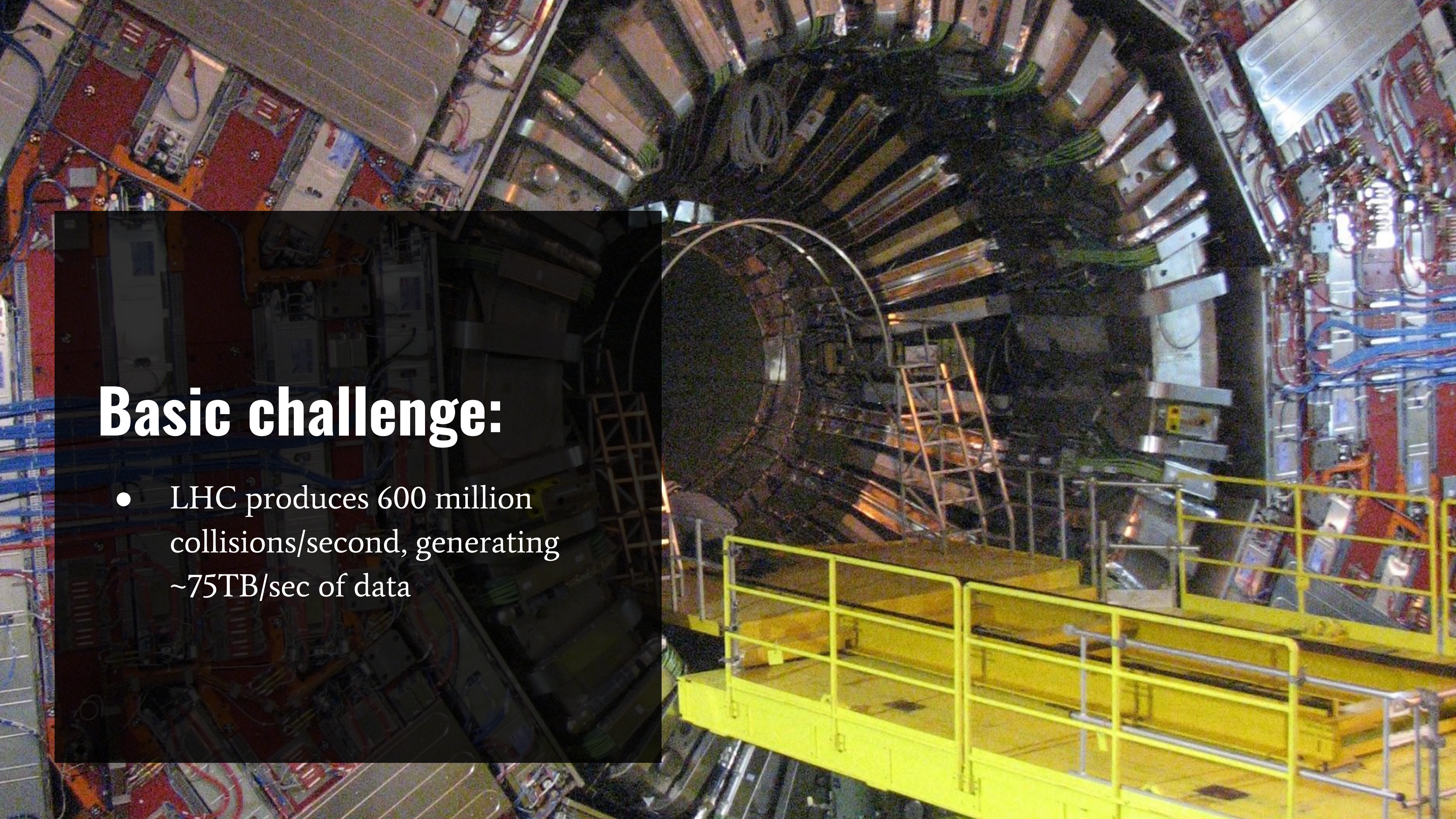
- The problem: Higgs detection at the Large Hadron Collider
- Quantum annealing overview
- Our technique: Quantum annealing for machine learning
- Results
- Future directions
- Acknowledgements

The problem:
Higgs detection at the Large Hadron Collider



The LHC:

- Large Hadron Collider -- 27km ring
- Cost: ~$4.5 billion



Basic challenge:

- LHC produces 600 million collisions/second, generating ~75TB/sec of data

Basic challenge:

- LHC produces 600 million collisions/second, generating ~75TB/sec of data
- Like the Biblical flood

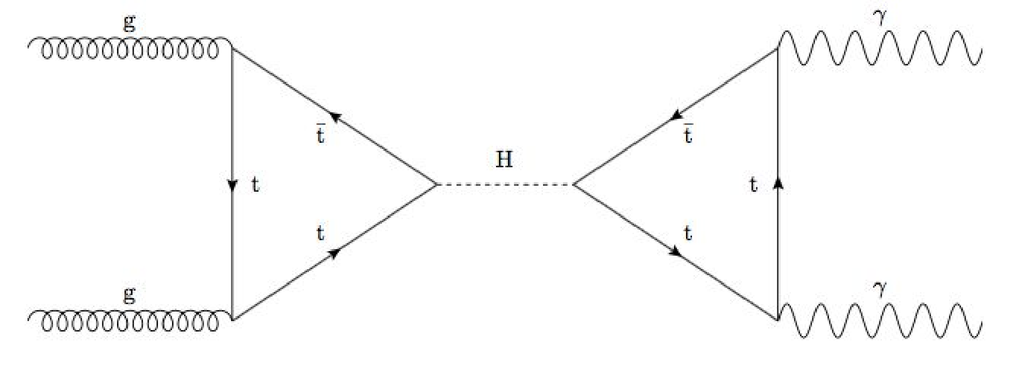
Basic challenge:

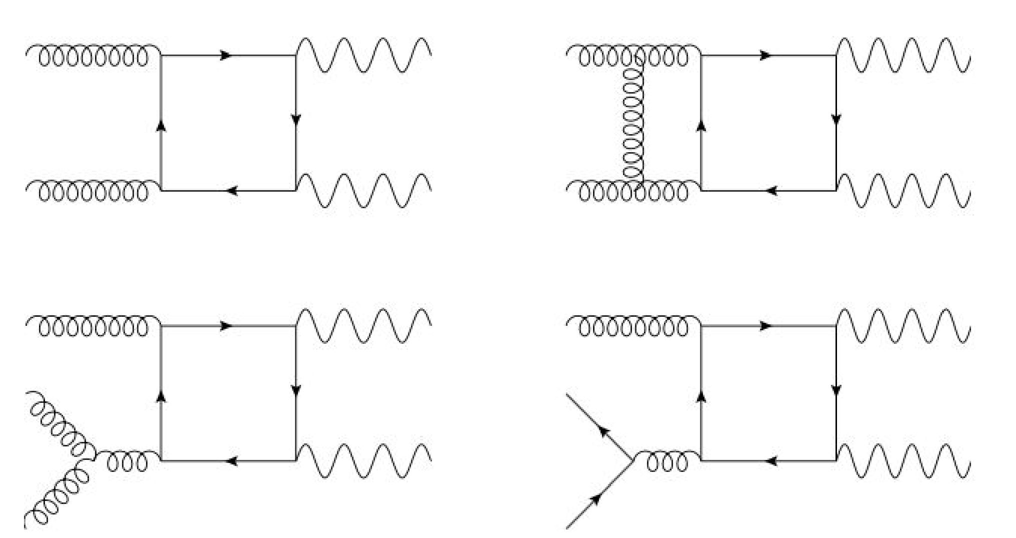
- LHC produces 600 million collisions/second, generating ~75TB/sec of data
- Like the Biblical flood
- Cut down to something closer to Niagara Falls, 1GB/sec of data

What process are we looking for anyway?

A Higgs decaying into two photons, i.e the $H \longrightarrow \gamma\gamma$ process

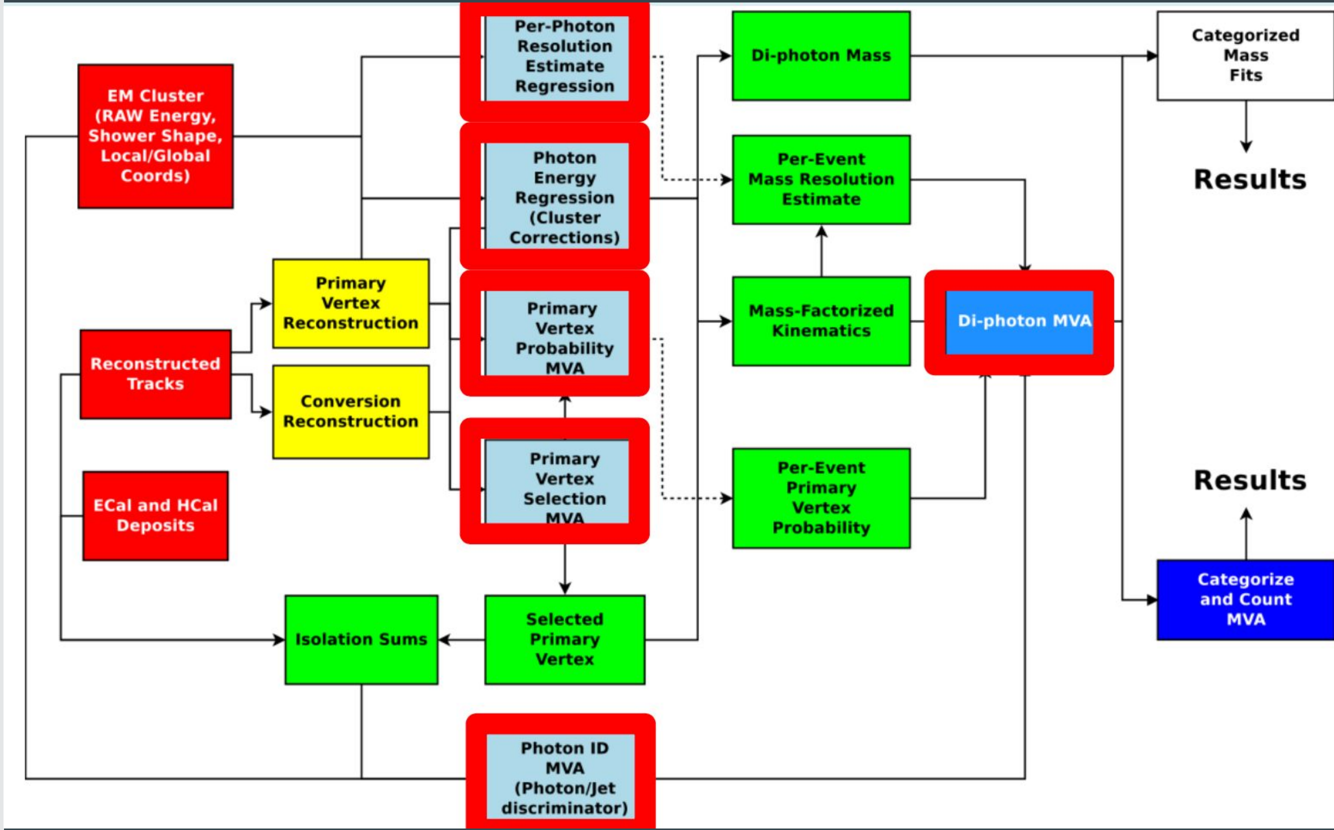
Background processes are, for instance, $gg \longrightarrow \gamma\gamma$ events





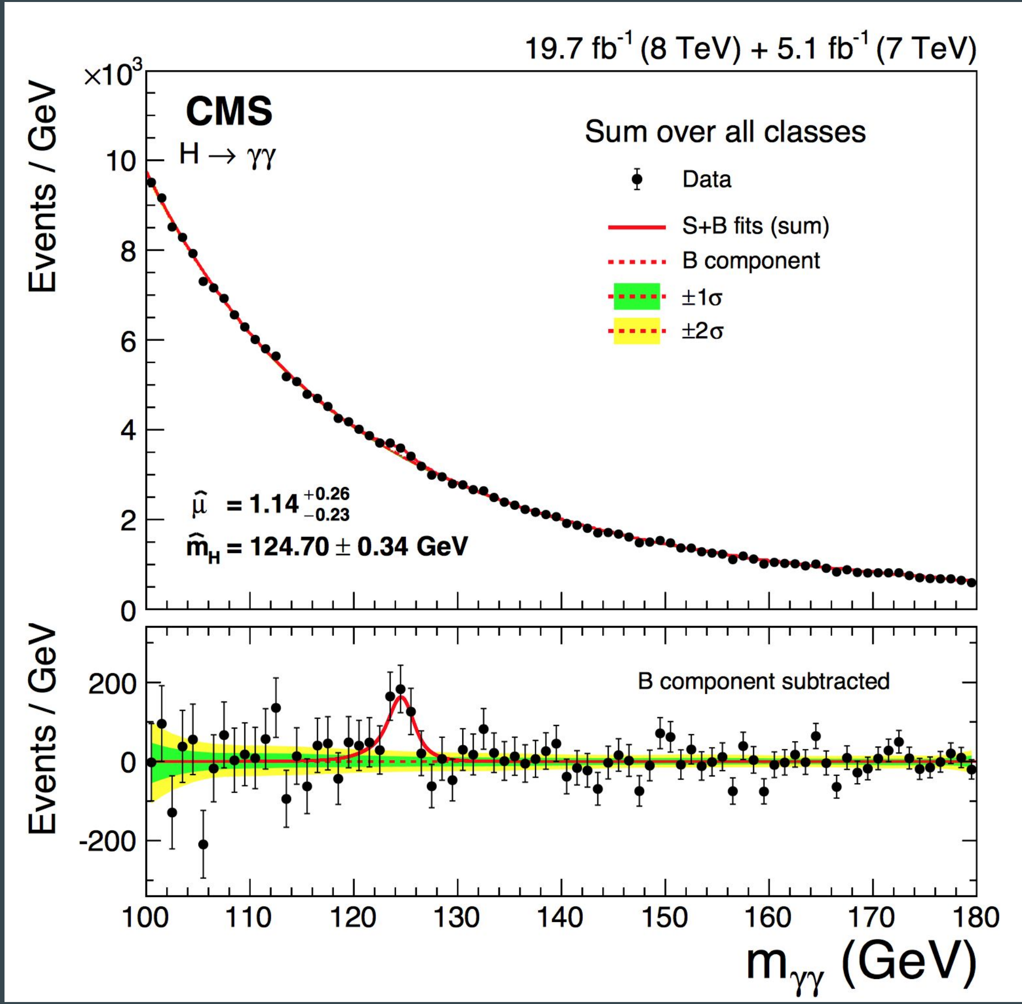
How they do it:

- Nested sets of triggers selecting the most interesting events according to criteria determined by simulations, discarding ~99.999% of the events
- May depend in part on boosted decision trees (BDTs) and multilayer perceptrons (MLPs aka neural nets, DNNs)



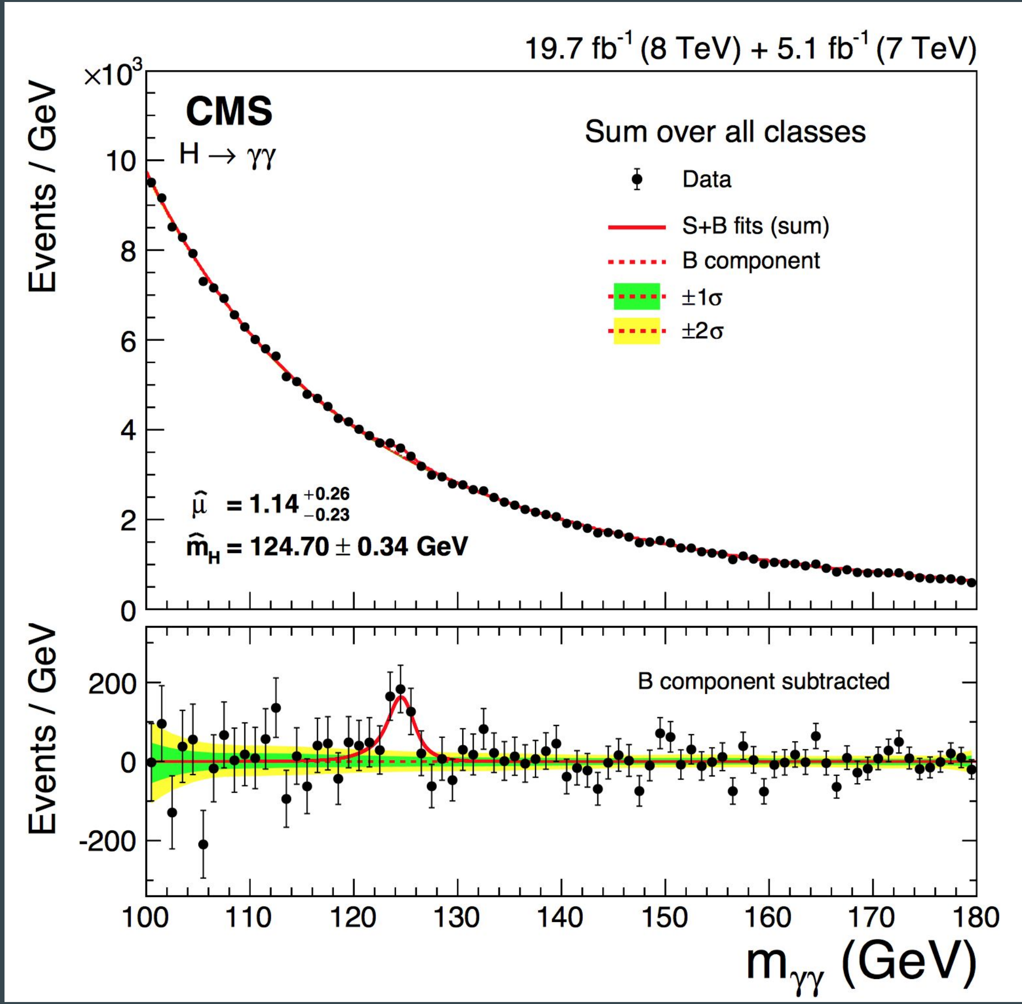
How they do it:

- Once you have a set of interesting events, you still have to classify which are signal (real Higgs decays, <5% of remaining events) and which background (other Standard Model processes, >95% of remaining events)
- Again typically using MLPs/DNNs or BDTs



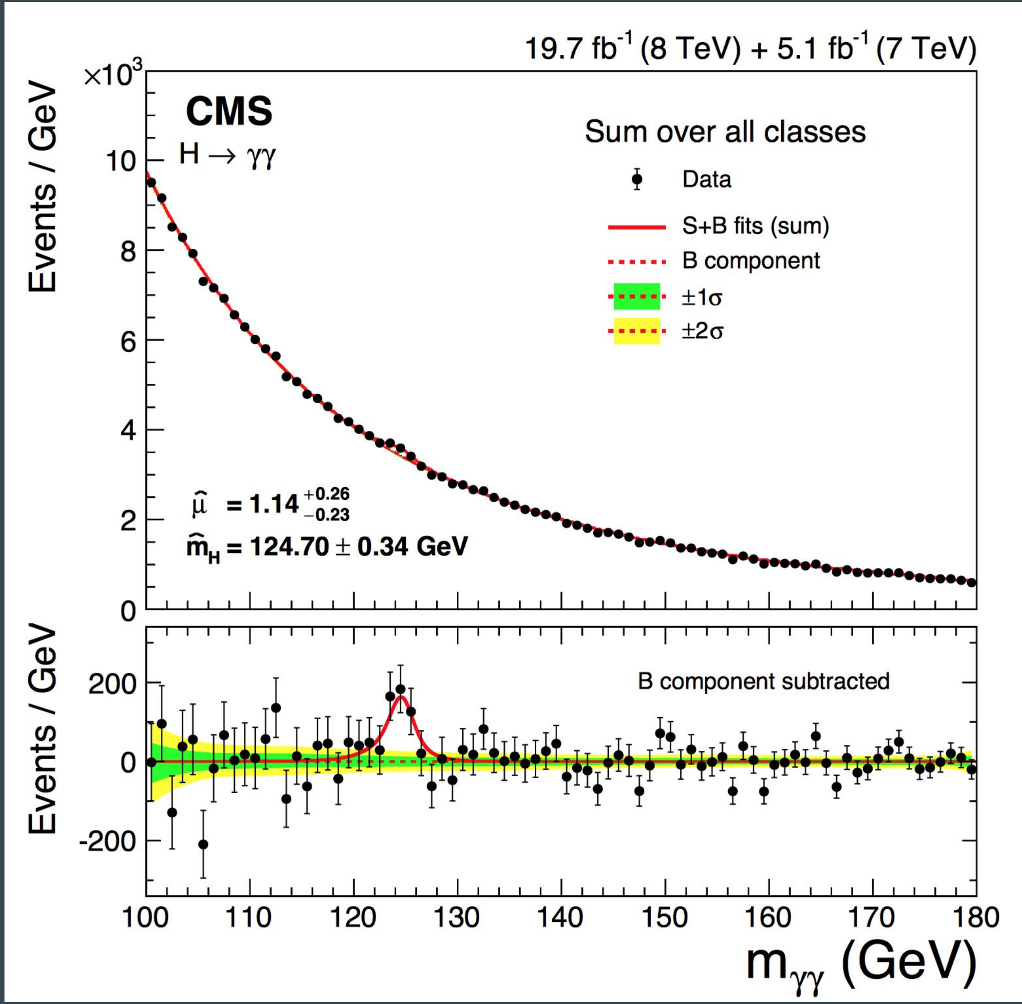
Challenges of BDTs/DNNs in this context:

- We don't have any real signal and background events
- Training data is all from simulated data from event generators which, while generally accurate, can't be fully trusted, and are more likely to be incorrect in the very high-level correlations BDTs and DNNs typically employ.



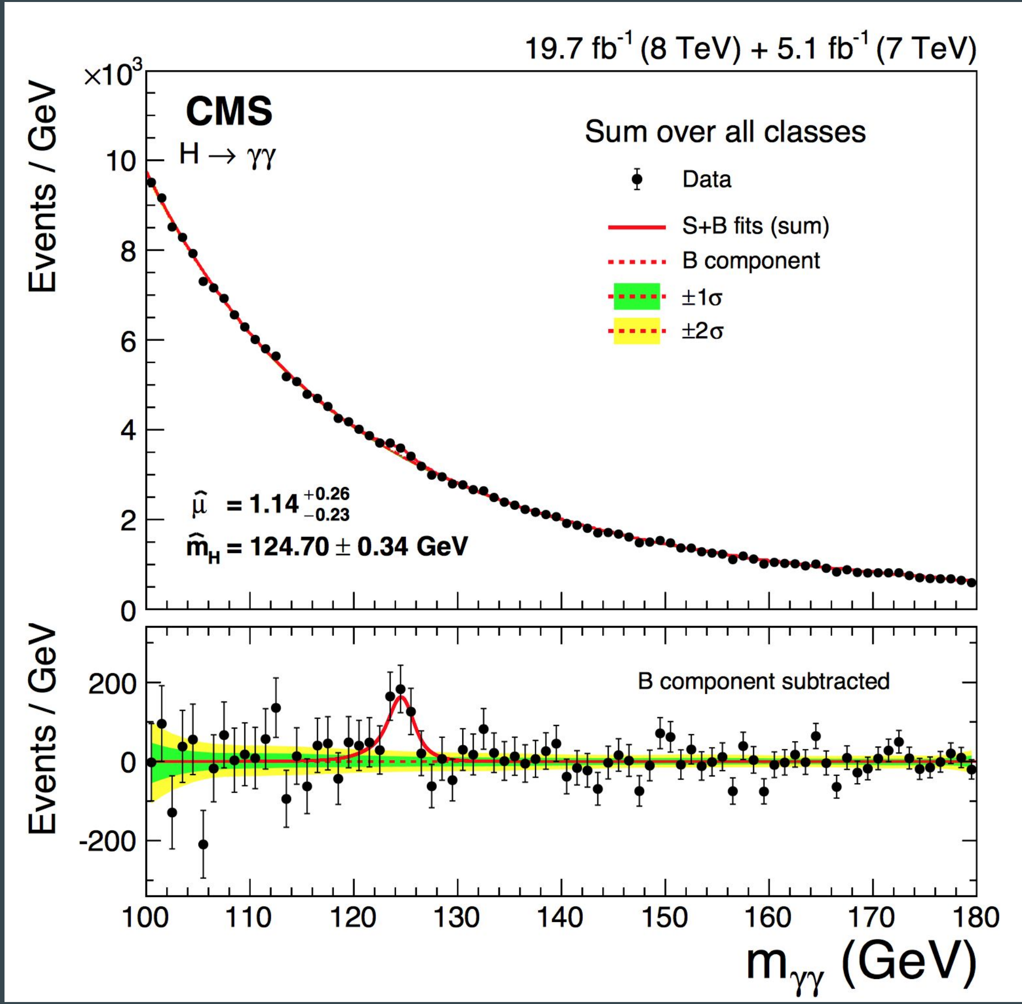
Challenges of BDTs/MLPs in this context:

- 2nd issue: interpretability
- MLPs are notoriously like black boxes, and while advances have been made in interpretation, still not easy to understand. BDTs are better but still nontrivial
- Would be better if we could directly interpret how it works and/or it gave us info about the important physics



Challenges of BDTs/MLPs in this context:

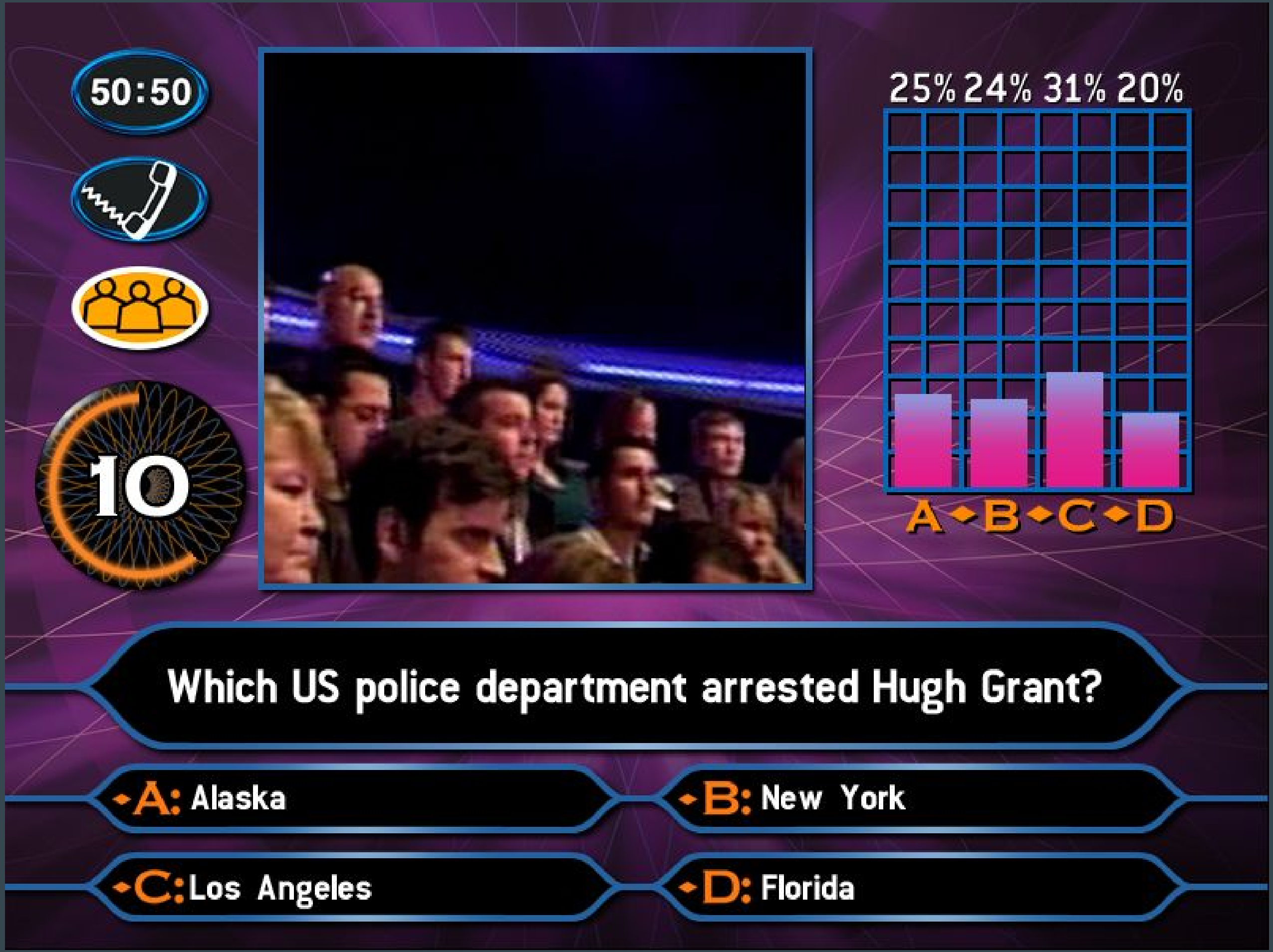
- Is there a potentially lighter, faster, more robust to simulation error, and/or more interpretable method we could use?
- Are there seemingly dead-end avenues that are opened up by newly developed special-purpose hardware, such as quantum annealers?



Our approach: QAML
Quantum annealing for machine learning

Basic idea: boosting

- Idea: if each person has a (very) rough idea of what the correct answer is, then polling many people will give a pretty good guess
- Given a set of weak classifiers, each only slightly better than random guessing, you construct a strong classifier by combining their output



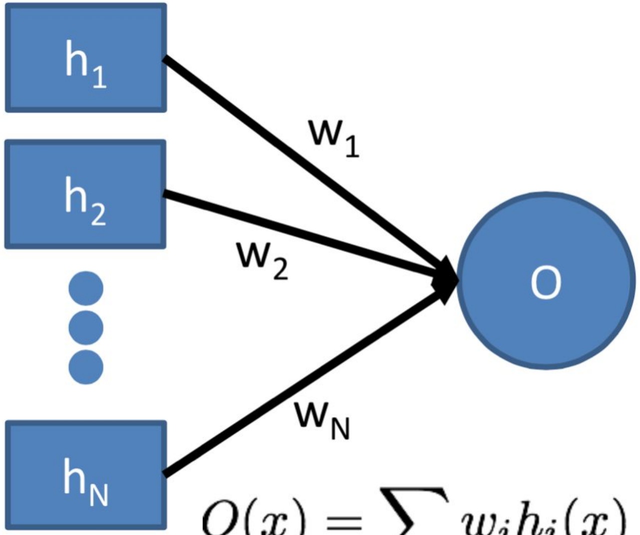
Boosting a binary classification

- $\mathcal{T} = \{(\vec{x}_i, y_i)\}$ is the training set, where $\vec{x}_i$ is some set of variables in the training set $y(x) = \begin{cases} +1, & \text{if } \in S, \\ -1, & \text{if } \in B \end{cases}$

- We define a weak classifier $h(\vec{x_i})$ such that

$$p(y_i = +1 | h(\vec{x_i}) > p(y_i = +1) > 0$$

$$-1 \leq h(\vec{x_i}) \leq 1 \forall \vec{x_i}$$



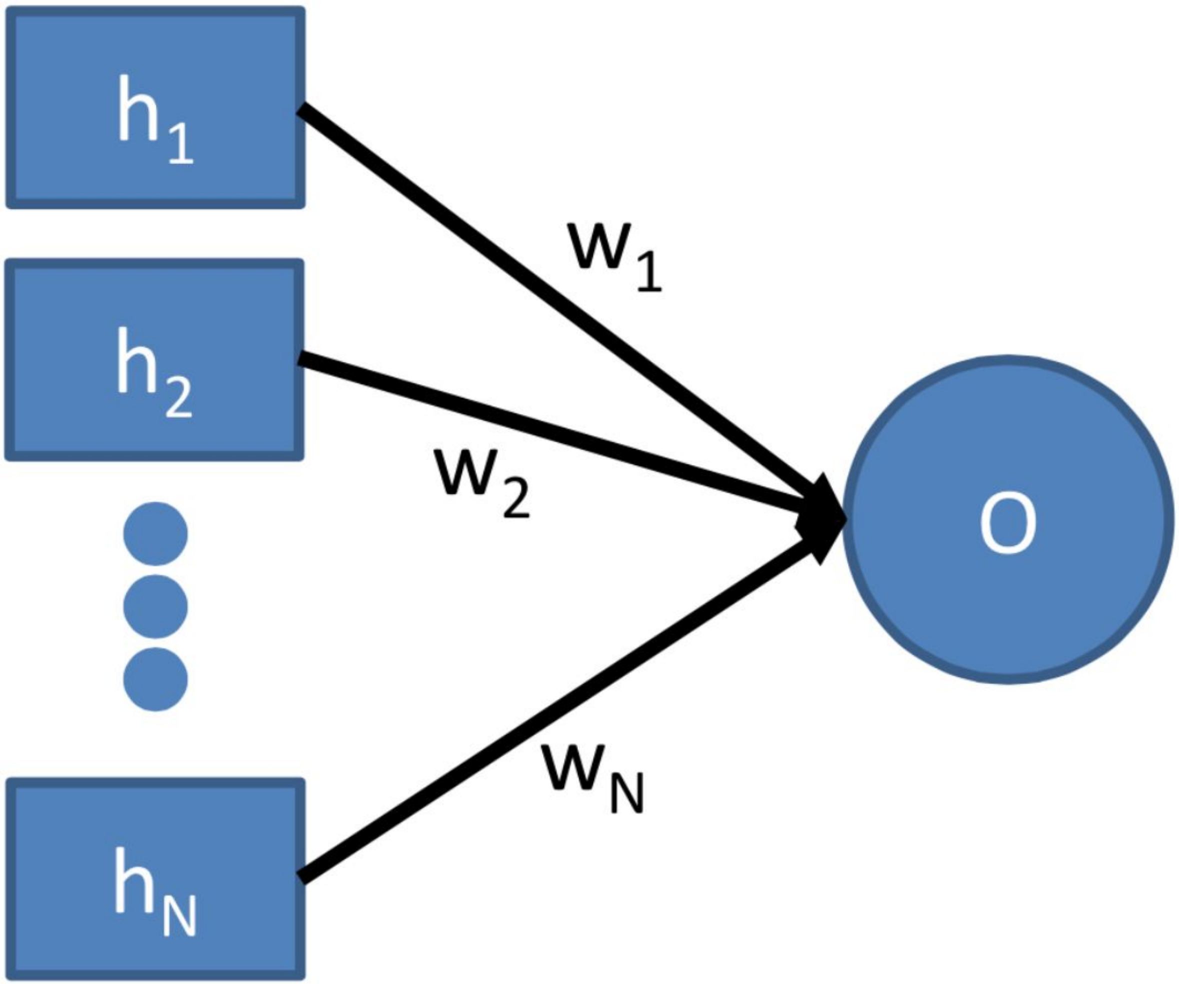
$$O(x) = \sum_i w_i h_i(x)$$

$$O(x) < O_c \rightarrow B$$

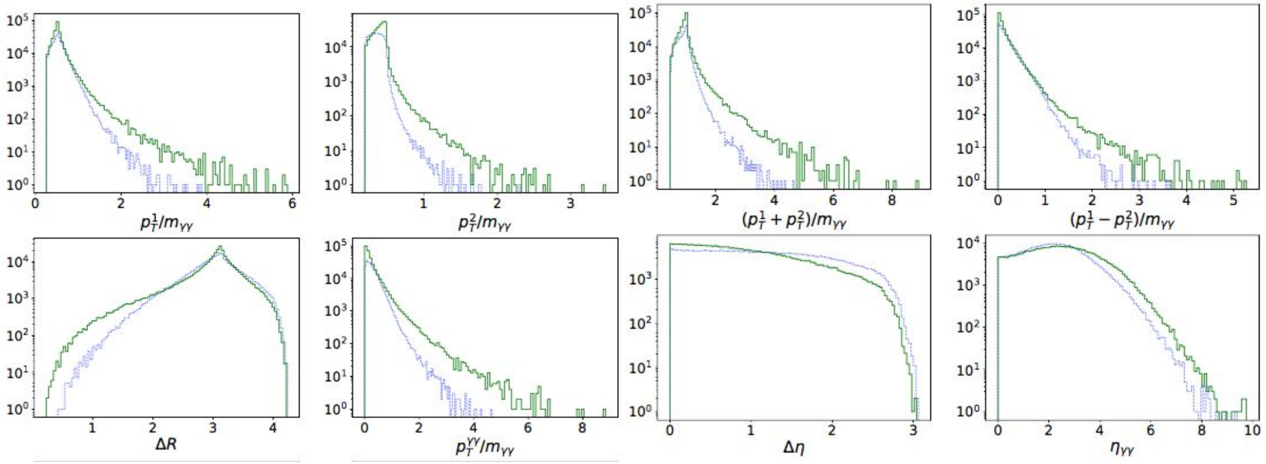
$$O(x) > O_c \rightarrow S$$

Weak classifiers

- In principle, can take any form, so long as it meets the aforementioned criteria
- What about our case?
- We're going to build weak classifiers using a reduced representation of the distribution over kinematic variables.
- What *are* said variables?



Our basic kinematic variables



variable	**description**
$p_T^1/m_{\gamma\gamma}$	transverse momentum of the highest p_T photon divided by the invariant mass of the diphoton pair
$p_T^2/m_{\gamma\gamma}$	transverse momentum of the second-highest p_T photon divided by the invariant mass of the diphoton pair
$(p_T^1+p_T^2)/m_{\gamma\gamma}$	sum of the transverse momentum of the two photons divided by their invariant mass
$(p_T^1-p_T^2)/m_{\gamma\gamma}$	difference of the transverse momentum of the two photons divided by their invariant mass
$p_T^{\gamma\gamma}/m_{\gamma\gamma}$	transverse momentum of the diphoton system divided by its invariant mass
$\Delta\eta$	difference in $\eta = -\log\tan\left(\frac{\theta}{2}\right)$, where θ is the angle with the beam axis
ΔR	sum in quadrature of the separation of and ϕ, the azimuthal angle of the two photons $(\sqrt{\Delta\eta^2+\Delta\phi^2})$
$\|\eta^{\gamma\gamma}\|$	the η value of the diphoton system

What do we want from our weak classifiers?

Interpretable/informative

Minimal sensitivity to errors in the event generators

Fast to evaluate (we're going to have many of them, so they can't be slow)

What do we want from our weak classifiers?

Interpretable/informative

Answer: Use only individual kinematic variables and their products/ratios, not higher-order correlations

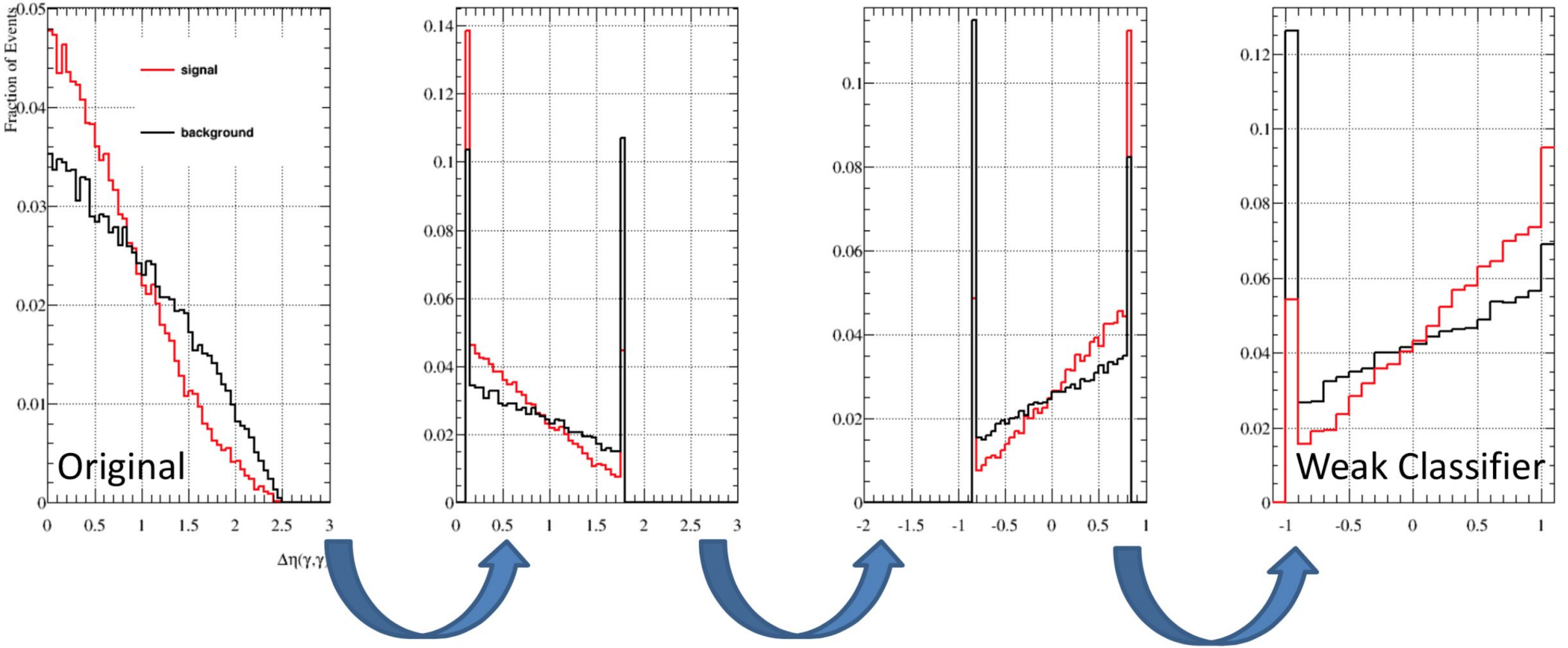
Minimal sensitivity to errors in the event generators

Answer: Ignore higher-order correlations, only use functions of certain quantiles of the distribution, neglect tails

Fast to evaluate (we're going to have many of them, so they can't be slow)

Answer: Use a linear function of a few quantiles

Construction of the Weak Classifiers



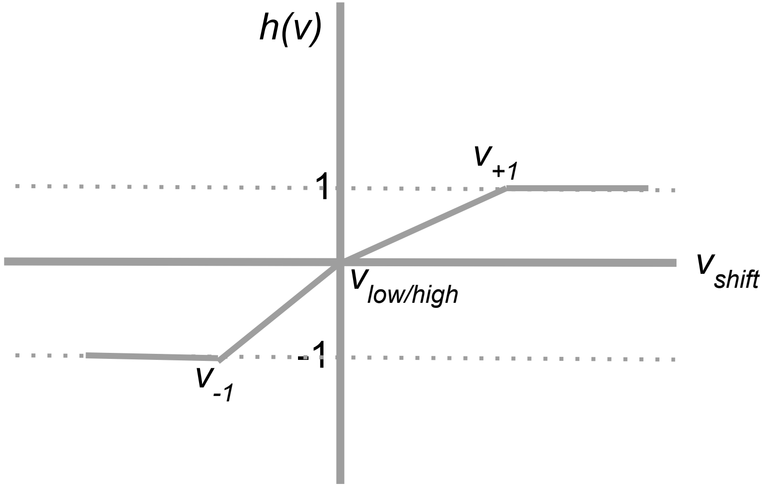
- Truncate Distributions
- Assign overflowing events to high/low bins

- Invert distribution
- Center at 0

- Rescale distribution so it fills [-1,1]

Math sketch:

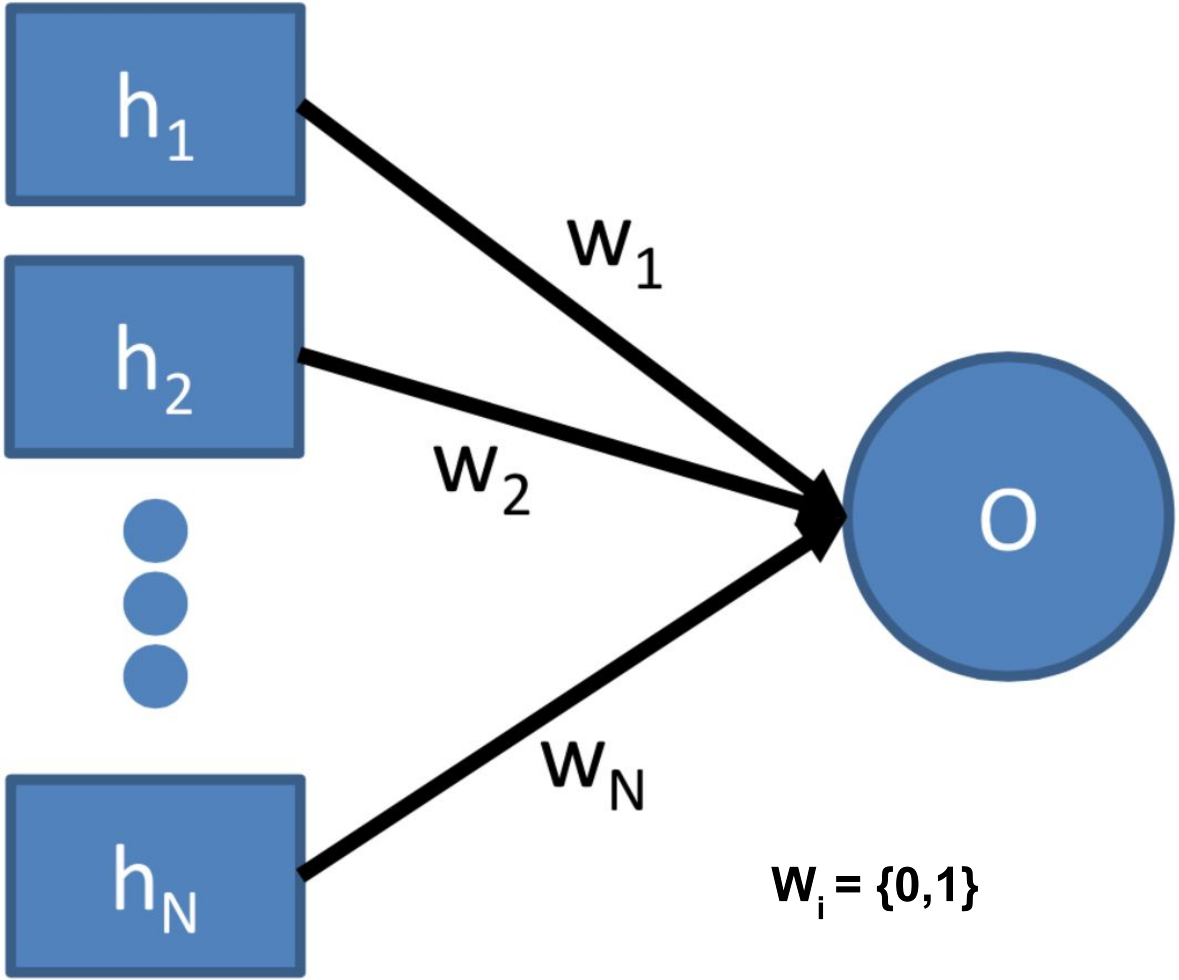
- S is the signal distribution, B background, v is the variable
- v_{low} and v_{high} are the 30th and 70th percentiles of S, b_{low} and b_{high} the percentiles on B at those values
- If $b_{high} < 0.7$ then define $v_{shift} = v_{low} - v$, elseif $b_{low} > 0.7$ then $v_{shift} = v - v_{high}$, else reject v
- Define v_{+1} and v_{-1} as the 10th and 90th percentile of the transformed S distribution
- With this formulation, the weak classifier is given by
- Do this for all the variables and products (or, if flipped flipped, the ratio)



$$h(v) = \begin{cases} +1 & \text{if } v_{+1} < v^{shift}(v) \\ \frac{v^{shift}(v)}{v_{+1}} & \text{if } 0 < v^{shift}(v) \leq v_{+1} \\ \frac{v^{shift}(v)}{|v_{-1}|} & \text{if } v_{-1} < v^{shift}(v) \leq 0 \\ -1 & \text{if } v^{shift}(v) < v_{-1} \end{cases}$$

Whither quantum annealing?

- So far, I haven't so much as mentioned quantum mechanics
- We're close though!
- The weights *w* haven't been restricted so far
- Let's choose to make them binary
  - Simpler optimization space as the weights are less sensitive to misspecification of *h*
  - Enables nice efficiency gains for optimization, ie conversion to a QUBO (quadratic, unconstrained binary optimization)



$W_i = \{0,1\}$

Constructing a QUBO problem

$$y(x) = \begin{cases} +1, & \text{if } \in S, \\ -1, & \text{if } \in B \end{cases} \quad O(x) = \sum_i w_i h_i(x)$$

$$\delta(\vec{w}) = \sum_{x \in T} \left(y(x) - \sum_i w_i h_i(x) \right)^2$$

per-event error: $E(x) = y(x) - \sum_{i=1}^{N} w_i h_i(x)$

$$\delta(\vec{w}) \propto \sum_{i,j} C_{ij} w_i w_j - 2 \sum_i C_{iy} w_i$$

total error: $\delta(\vec{w}) = \sum_{x \in T} E(x)^2$

Minimize:

$$\delta(\vec{w}) \propto \sum_{i,j} C_{ij} w_i w_j + \sum_i (\lambda - 2C_{iy}) w_i$$

What can you do with a QUBO?

Run simulated annealing or parallel tempering algorithms (fully classical)

Submit the problem to a quantum annealer to solve --- D-Wave QA processors solve QUBOs natively

Brief overview of quantum annealing

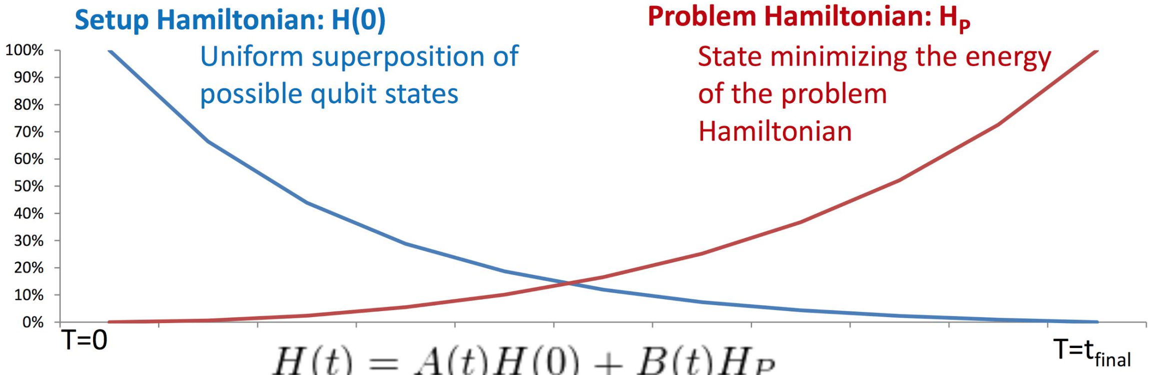
What is quantum annealing?

Roughly, one initializes a system of two-state quantum systems (qubits), label the states {-1,+1}

Initialize the system in a trivial Hamiltonian H(0) and allow it to find the ground state

Slowly change the Hamiltonian, turning off H(0) and increasing the strength of target H_P until $H=H_P$

This final Hamiltonian corresponds to your QUBO problem



$$H(t) = A(t)H(0) + B(t)H_P$$

What is quantum annealing?

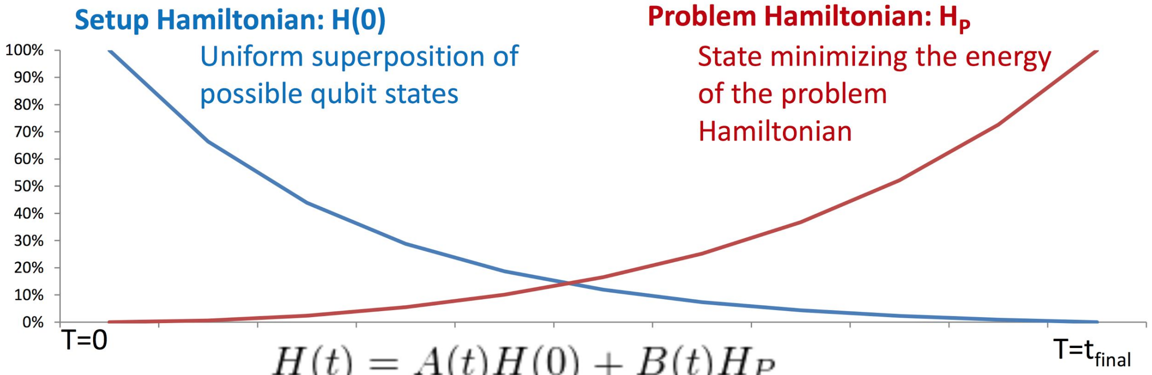
$$H(0) = \sum_i \sigma_i^x$$

$$H_P = \sum_{i,j} J_{ij} s_i s_j + \sum_i h_i s_i$$

H_P is effectively $\delta(\vec{w}) \propto \sum_{i,j} C_{ij} w_i w_j + \sum_i (\lambda - 2C_{iy}) w_i$

σ_i^x has a ground state of proportional to $|0\rangle + |1\rangle$

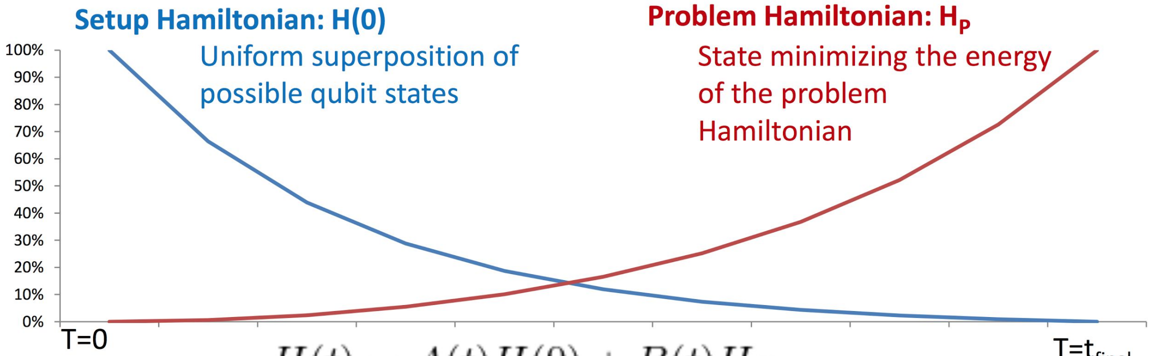
H(0) has no interactions, so cools to ground state quickly, and the total ground state is an equal superposition over all bitstrings



$$H(t) = A(t)H(0) + B(t)H_P$$

Why quantum annealing?

- Because we can
- We suspect that with an appropriately designed quantum annealer one can find the ground state more quickly via tunneling than one can through simple thermalization alone
- Hardware and algorithms are developing rapidly, with feedback between producers (to date, primarily D-Wave Systems) and users, so we could effect the future trajectory of development



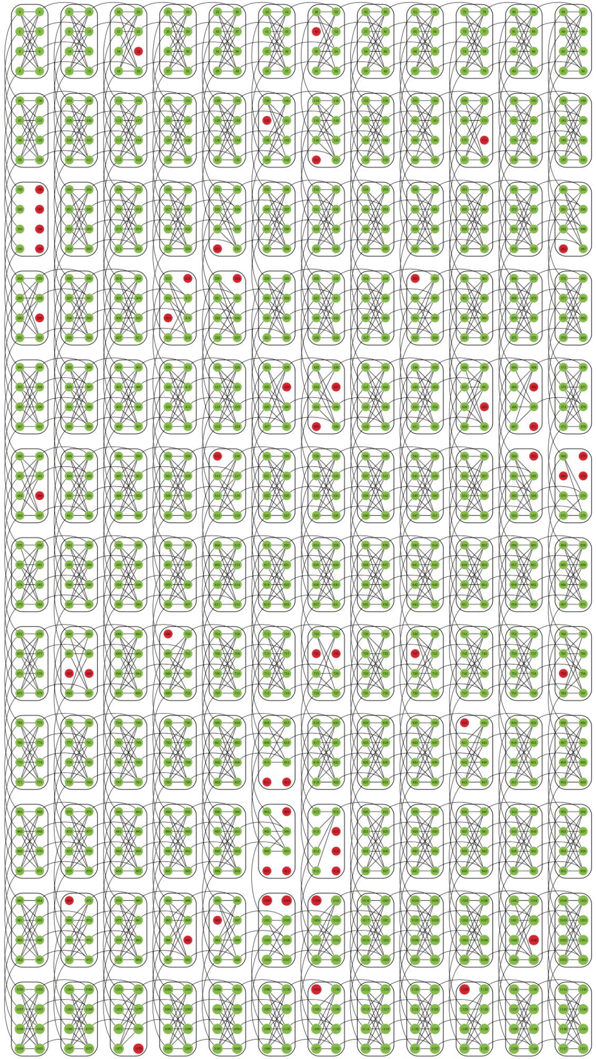
$$H(t) = A(t)H(0) + B(t)H_P$$

Our quantum annealer

- Built by D-Wave Systems in Burnaby, CA
- 1152 qubits nominal, 1098 functioning/active
- Chilled to 15mK

Hardware graph:

- Red are inactive qubits
- Lines are couplers
- Green are active qubits



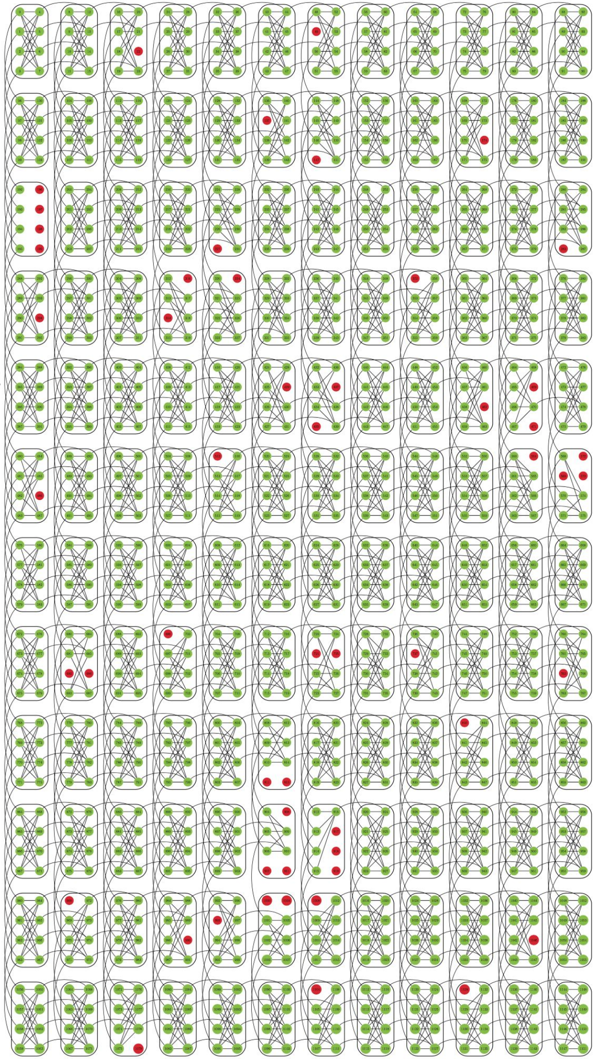
Our quantum annealer

- Not fully connect
- But our problem is minimizing

$$\delta(\vec{w}) \propto \sum_{i,j} C_{ij} w_i w_j + \sum_i (\lambda - 2C_{iy}) w_i$$

That's fully connected, the sum is over all *i,j*.

What to do…

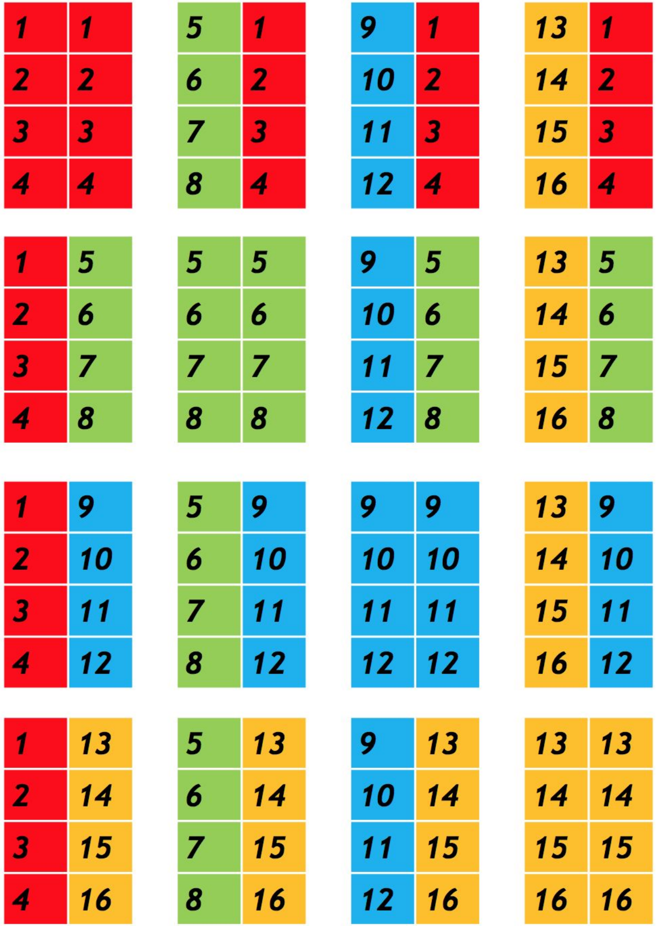


Minor embedding: When a chain feels like a qubit

Bind qubits in a chain together very tightly, with an energy that is J_F times stronger than the couplings of the problem

Split local field across all qubits in the chain

Decode states returned from the annealer by majority vote



Our problems:

- Training dataset is approximately 200k signal and background events (each), divided into 20 sets of 10k each to estimate random variation from dataset
- Testing set is approximately 100k events
- Signal data generated using 125 GeV Higgs decays produced by gluon fusion at 8TeV collisions using PYTHIA 6.4
- Background data of Standard Model processes generated using SHERPA after restricting to processes that meet realistic trigger and detector acceptance requirements, $p^T_1 > 32$ GeV, $p^T_2 > 25$ GeV with diphoton mass 122.5 GeV $< m_{\gamma\gamma} <$ 127.5 GeV and $|\eta|<2.5$
- Used training sizes of 100, 1000, 5000, 10k, 15k, and 20k events, 20 such sets per size, and split evenly between signal and background

Our problems:

- Training dataset is approximately 200k signal and background events (each), divided into 20 sets of 10k each to estimate random variation from dataset
- Testing set is approximately 100k events
- Signal data generated using 125 GeV Higgs decays produced by gluon fusion at 8TeV collisions using PYTHIA 6.4
- Background data of Standard Model processes generated using SHERPA after restricting to processes that meet realistic trigger and detector acceptance requirements, $p^T_1 > 32$ GeV, $p^T_2 > 25$ GeV with diphoton mass 122.5 GeV $< m_{\gamma\gamma} <$ 127.5 GeV and $|\eta|<2.5$
- Used training sizes of 100, 1000, 5000, 10k, 15k, and 20k events, 20 such sets per size, and split evenly between signal and background

Results, at long last

Physical insight

λ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	20	20	20	19	20	20	0	5	20	20	20	20	19	20	17	20	20
0.01	0	20	20	20	19	20	20	0	4	20	20	20	20	19	20	17	20	20
0.02	0	20	20	20	19	20	20	0	4	20	20	20	20	19	20	16	20	20
0.05	0	20	20	20	19	20	20	0	1	20	20	20	20	19	20	10	20	17
0.1	0	20	20	20	19	20	20	0	0	20	20	20	20	19	20	6	14	2
0.2	0	20	20	20	19	20	20	0	0	20	14	20	20	12	20	4	1	0
0.4	0	20	0	2	19	20	20	0	0	20	17	20	20	0	20	1	0	0
0.8	0	20	0	0	0	0	9	0	0	18	0	0	20	0	2	0	0	0

λ	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	20	0	0	19	0	20	0	3	0	20	19	7	0	15	0	19	20	20
0.01	20	0	0	19	0	20	0	2	0	20	19	6	0	15	0	19	20	20
0.02	20	0	0	19	0	20	0	1	0	20	19	4	0	15	0	19	20	20
0.05	20	0	0	19	0	20	0	0	0	20	16	1	0	11	0	19	20	20
0.1	20	0	0	1	0	20	0	0	0	20	1	0	0	5	0	16	20	20
0.2	18	0	0	0	0	20	0	0	0	20	0	0	0	0	0	0	20	20
0.4	0	0	0	0	0	7	0	0	0	20	0	0	0	0	0	0	20	3
0.8	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	19	0

1	2	3	4	5	6	7	8	9
p_T^1	p_T^2	ΔR	$p_T^{\gamma\gamma}$	$p_T^1+p_T^2$	$p_T^1-p_T^2$	$\Delta\eta$	$\eta_{\gamma\gamma}$	$(p_T^1+p_T^2)\eta_{\gamma\gamma}$
10	11	12	13	14	15	16	17	18
$\frac{p_T^2}{p_T^1-p_T^2}$	$\frac{p_T^2}{\Delta\eta}$	$p_T^2\eta_{\gamma\gamma}$	$\frac{1}{\Delta R p_T^{\gamma\gamma}}$	$\frac{p_T^1+p_T^2}{\Delta R}$	$\frac{1}{\Delta R(p_T^1-p_T^2)}$	$\frac{1}{\Delta R\Delta\eta}$	$\frac{\eta_{\gamma\gamma}}{\Delta R}$	$\frac{1}{(p_T^1-p_T^2)\Delta\eta}$
19	20	21	22	23	24	25	26	27
$p_T^1 p_T^2$	$\frac{p_T^1}{\Delta R}$	$\frac{p_T^1}{p_T^{\gamma\gamma}}$	$p_T^1(p_T^1+p_T^2)$	$\frac{p_T^1}{p_T^1-p_T^2}$	$\frac{p_T^1}{\Delta\eta}$	$\frac{p_T^1}{\eta_{\gamma\gamma}}$	$\frac{p_T^2}{\Delta R}$	$\frac{\eta_{\gamma\gamma}}{p_T^1-p_T^2}$
28	29	30	31	32	33	34	35	36
$\frac{p_T^2}{p_T^{\gamma\gamma}}$	$p_T^2(p_T^1+p_T^2)$	$\frac{p_T^1+p_T^2}{p_T^{\gamma\gamma}}$	$\frac{\eta_{\gamma\gamma}}{p_T^{\gamma\gamma}}$	$\frac{1}{p_T^{\gamma\gamma}\Delta\eta}$	$\frac{1}{p_T^{\gamma\gamma}(p_T^1-p_T^2)}$	$\frac{p_T^1+p_T^2}{p_T^1-p_T^2}$	$\frac{p_T^1+p_T^2}{\Delta\eta}$	$\frac{\eta_{\gamma\gamma}}{\Delta\eta}$

Physical insight

λ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	20	20	20	19	20	20	0	5	20	20	20	20	19	20	17	20	20
0.01	0	20	20	20	19	20	20	0	4	20	20	20	20	19	20	17	20	20
0.02	0	20	20	20	19	20	20	0	4	20	20	20	20	19	20	16	20	20
0.05	0	20	20	20	19	20	20	0	1	20	20	20	20	19	20	10	20	17
0.1	0	20	20	20	19	20	20	0	0	20	20	20	20	19	20	6	14	2
0.2	0	20	20	20	19	20	20	0	0	20	14	20	20	12	20	4	1	0
0.4	0	20	0	2	19	20	20	0	0	20	17	20	20	0	20	1	0	0
0.8	0	20	0	0	0	0	9	0	0	18	0	0	20	0	2	0	0	0

λ	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	20	0	0	19	0	20	0	3	0	20	19	7	0	15	0	19	20	20
0.01	20	0	0	19	0	20	0	2	0	20	19	6	0	15	0	19	20	20
0.02	20	0	0	19	0	20	0	1	0	20	19	4	0	15	0	19	20	20
0.05	20	0	0	19	0	20	0	0	0	20	16	1	0	11	0	19	20	20
0.1	20	0	0	1	0	20	0	0	0	20	1	0	0	5	0	16	20	20
0.2	18	0	0	0	0	20	0	0	0	20	0	0	0	0	0	0	20	20
0.4	0	0	0	0	0	7	0	0	0	20	0	0	0	0	0	0	20	3
0.8	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	19	0

20k training events, number of problems (out of 20) where variable is active in the ground state configuration of the Hamiltonian (the ideal solution)

Three variables survive for extremely high regularization strength λ, $[p_T^2, (\Delta R p_T^{\gamma\gamma})^{-1}, \text{ and } \frac{p_T^2}{p_T^{\gamma\gamma}}]$

Physical insight

λ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	20	20	20	19	20	20	0	5	20	20	20	20	19	20	17	20	20
0.01	0	20	20	20	19	20	20	0	4	20	20	20	20	19	20	17	20	20
0.02	0	20	20	20	19	20	20	0	4	20	20	20	20	19	20	16	20	20
0.05	0	20	20	20	19	20	20	0	1	20	20	20	20	19	20	10	20	17
0.1	0	20	20	20	19	20	20	0	0	20	20	20	20	19	20	6	14	2
0.2	0	20	20	20	19	20	20	0	0	20	14	20	20	12	20	4	1	0
0.4	0	20	0	2	19	20	20	0	0	20	17	20	20	0	20	1	0	0
0.8	0	20	0	0	0	0	9	0	0	18	0	0	20	0	2	0	0	0

λ	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	20	0	0	19	0	20	0	3	0	20	19	7	0	15	0	19	20	20
0.01	20	0	0	19	0	20	0	2	0	20	19	6	0	15	0	19	20	20
0.02	20	0	0	19	0	20	0	1	0	20	19	4	0	15	0	19	20	20
0.05	20	0	0	19	0	20	0	0	0	20	16	1	0	11	0	19	20	20
0.1	20	0	0	1	0	20	0	0	0	20	1	0	0	5	0	16	20	20
0.2	18	0	0	0	0	20	0	0	0	20	0	0	0	0	0	0	20	20
0.4	0	0	0	0	0	7	0	0	0	20	0	0	0	0	0	0	20	3
0.8	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	19	0

Why are they the strongest? The major difference between signal and background is the creation of a heavy particle, the Higgs. It takes a lot of energy to boost perpendicular to the beam axis, so Higgs events likely have smaller transverse momentum $p_{\gamma\gamma}^{T}$, and for this to be correlated with the angle to the beam axis, which is part of ΔR.

Physical insight

λ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	20	20	20	19	20	20	0	5	20	20	20	20	19	20	17	20	20
0.01	0	20	20	20	19	20	20	0	4	20	20	20	20	19	20	17	20	20
0.02	0	20	20	20	19	20	20	0	4	20	20	20	20	19	20	16	20	20
0.05	0	20	20	20	19	20	20	0	1	20	20	20	20	19	20	10	20	17
0.1	0	20	20	20	19	20	20	0	0	20	20	20	20	19	20	6	14	2
0.2	0	20	20	20	19	20	20	0	0	20	14	20	20	12	20	4	1	0
0.4	0	20	0	2	19	20	20	0	0	20	17	20	20	0	20	1	0	0
0.8	0	20	0	0	0	0	9	0	0	18	0	0	20	0	2	0	0	0

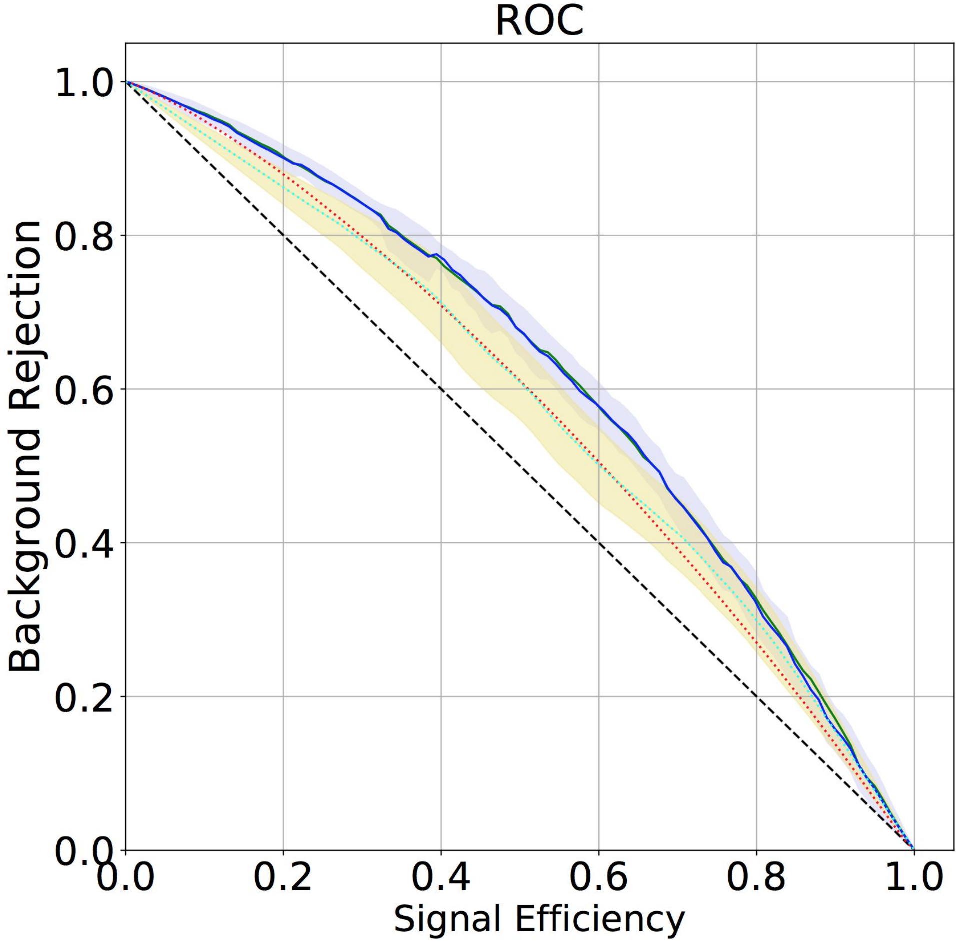
λ	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	20	0	0	19	0	20	0	3	0	20	19	7	0	15	0	19	20	20
0.01	20	0	0	19	0	20	0	2	0	20	19	6	0	15	0	19	20	20
0.02	20	0	0	19	0	20	0	1	0	20	19	4	0	15	0	19	20	20
0.05	20	0	0	19	0	20	0	0	0	20	16	1	0	11	0	19	20	20
0.1	20	0	0	1	0	20	0	0	0	20	1	0	0	5	0	16	20	20
0.2	18	0	0	0	0	20	0	0	0	20	0	0	0	0	0	0	20	20
0.4	0	0	0	0	0	7	0	0	0	20	0	0	0	0	0	0	20	3
0.8	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	19	0

Similarly, with less transverse momentum we expect the two photons to have similar momenta, and thus p^T_2 will be larger than typical background events and to be a larger fraction of the total diphoton momentum than typical.

Good luck tweaking a neural net or random forest and having it lead you toward understanding the physics!

ROC curves

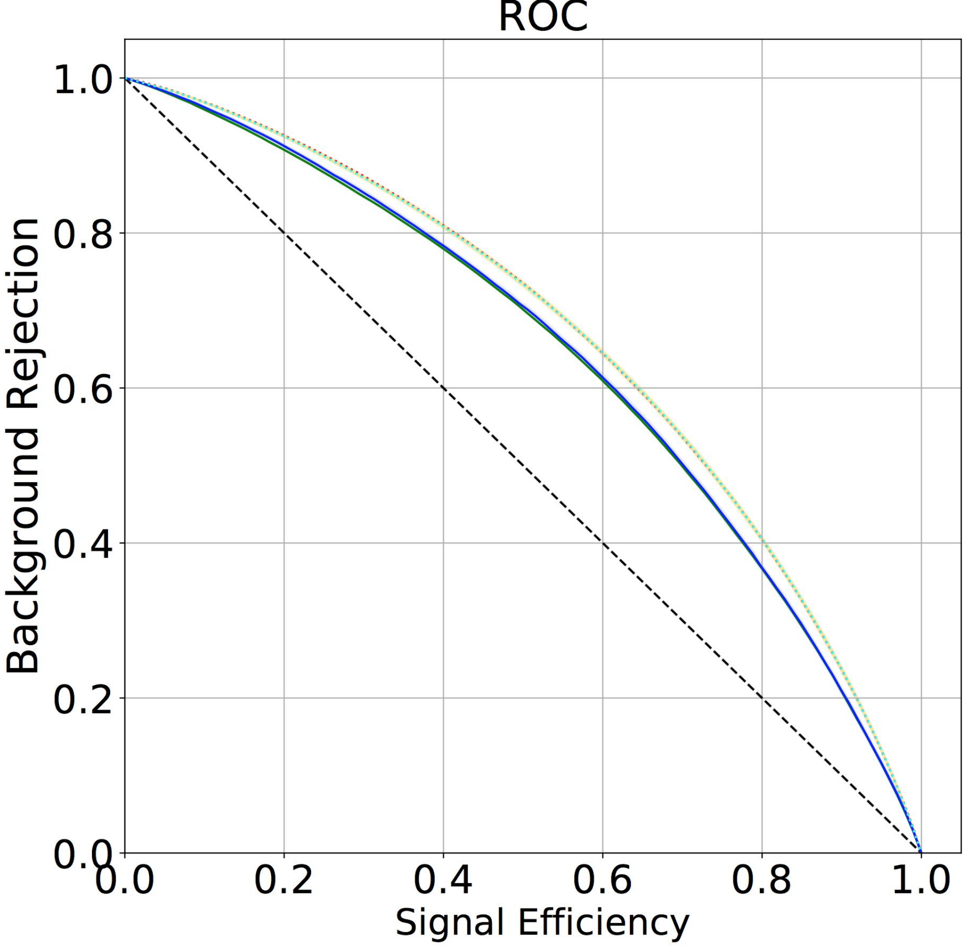
Color key:
D-Wave (DW) - green
Simulated annealing (SA) - blue
XGBoost (XGB, decision trees) - cyan
Deep Neural Net (DNN) - red



100 training events

ROC curves

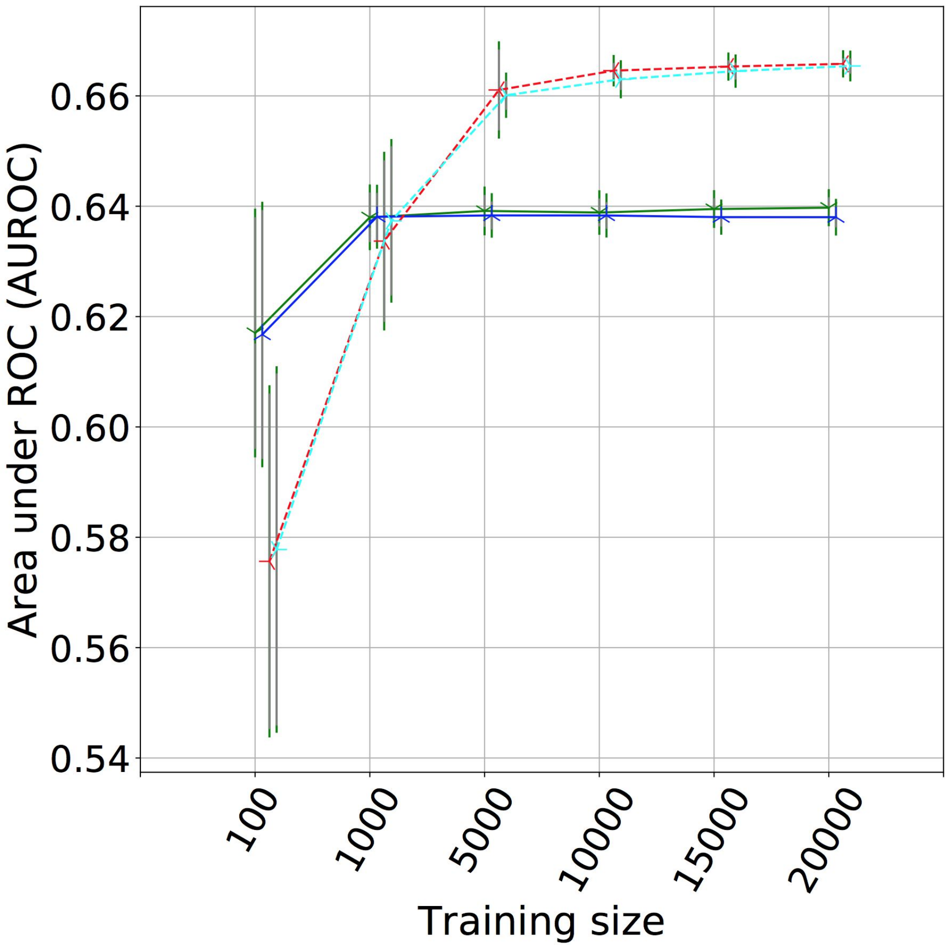
Color key:
D-Wave (DW) - green
Simulated annealing (SA) - blue
XGBoost (XGB, decision trees) - cyan
Deep Neural Net (DNN) - red



20k training events

AUROC curves

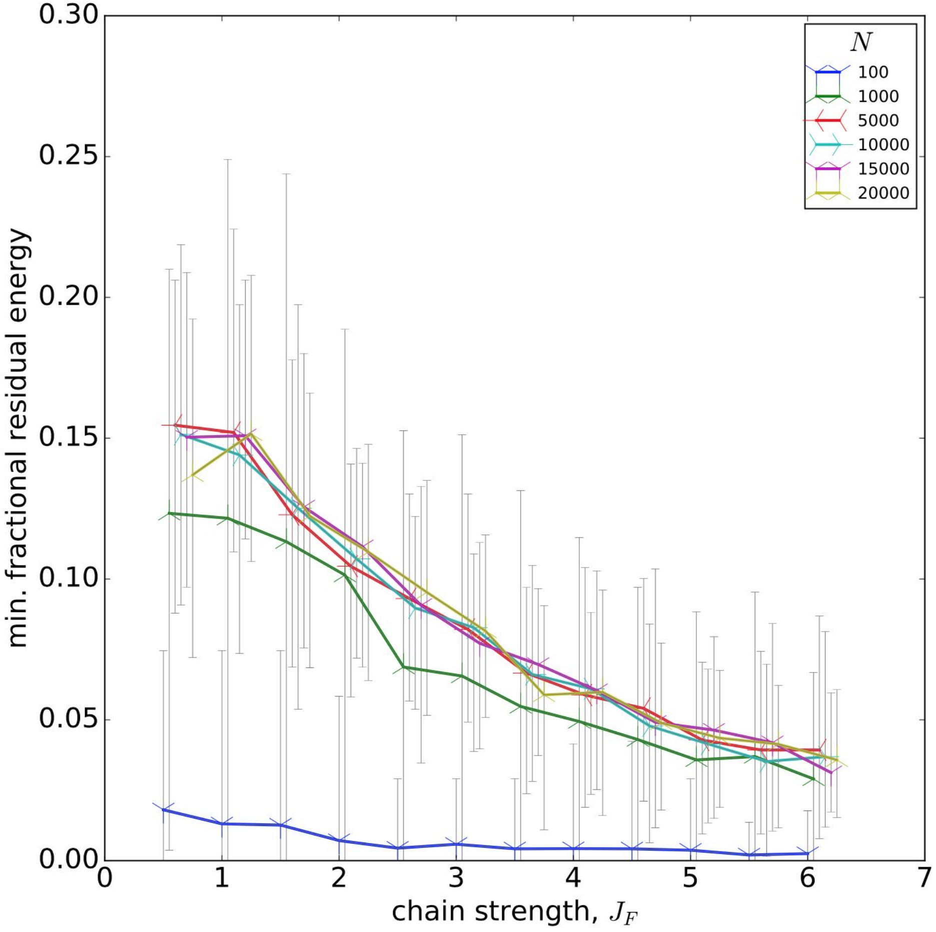
Color key:
D-Wave (DW) - green
Simulated annealing (SA) - blue
XGBoost (XGB, decision trees) - cyan
Deep Neural Net (DNN) - red



20k training events

Why does SA perform a bit better than DW?

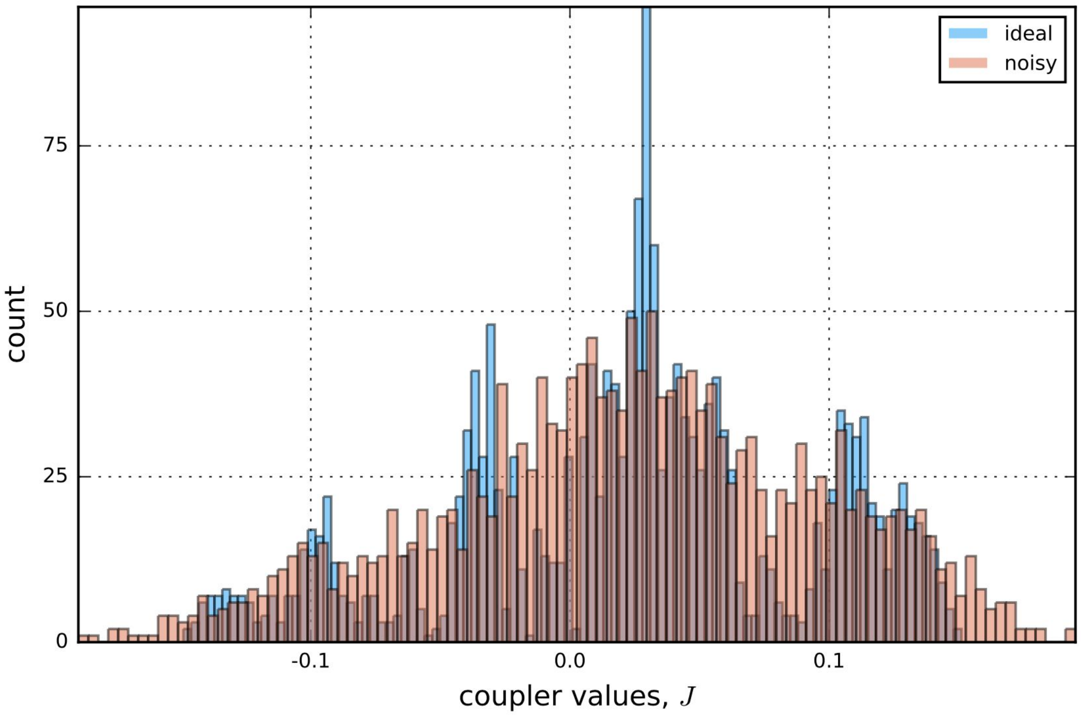
Broken chains
DW has them, SA doesn't



Why does SA perform a bit better than DW?

Also noise:
SA runs on logical problem with floating point precision

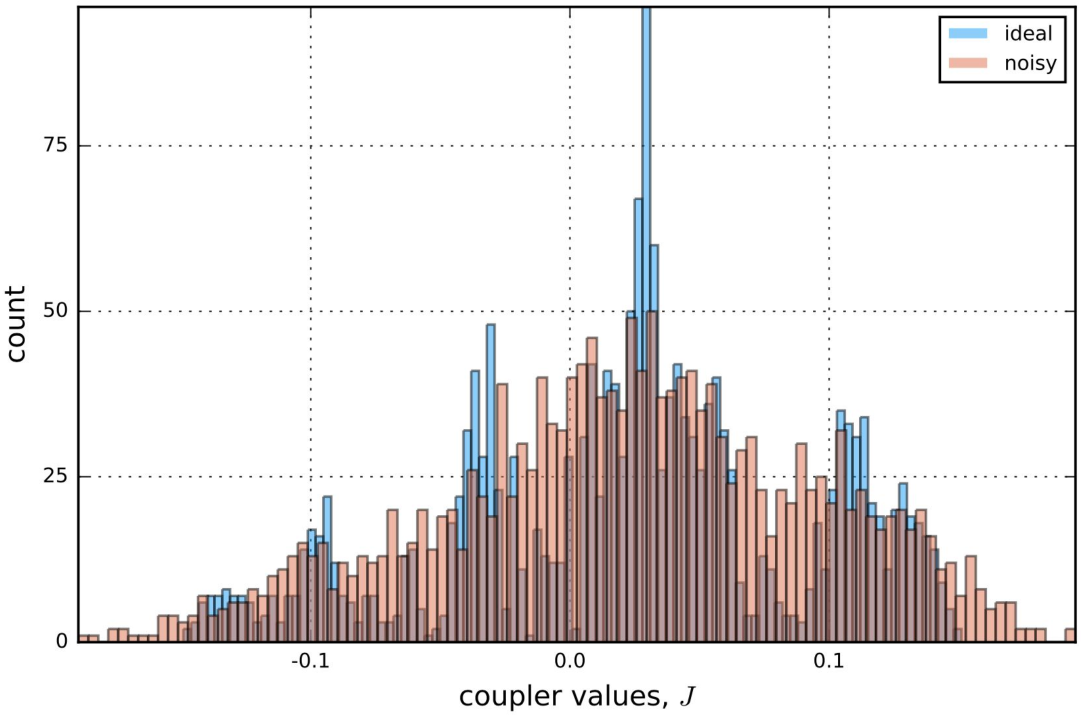
DW runs on hardware with errors of ~3%



Why does SA perform a bit better than DW?

Both problems are being addressed in future quantum annealers

- More couplings = shorter chains = fewer broken qubits
- Stronger couplings = fewer broken chains
- Lower noise on couplings



Where can we go with this?

- QAML can be run on classical hardware as well as quantum, enabling tests for larger and more difficult problems, more complex decay processes, etc.
- Continuing advances in quantum annealers should enable significant improvements in their performance, and so should likely stay competitive or exceed classical solvers for QAML
- More advanced procedures:
  - Some variables dominate, and is obvious from solutions, we could pin them to their value, simplify the Hamiltonian, cut the number of needed qubits, and thereby improve QA/DW's capacity to find the ground state configuration
  - Error correction and MAB techniques to improve solutions from DW
  - Use QAML for triggers -- fast/simple, reasonably accurate at small samples
  - New variants for weak classifiers
  - Quantum boltzmann machines -- very different, but promising

QAML outperforms standard methods for small sizes, is robust to generator error, highly interpretable, and readily implementable on quantum and physical annealers.

Thanks!

This project is supported in part by the United States Department of Energy, Office of High Energy Physics Research Technology Computational HEP and Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359. The project is also supported in part under ARO grant number W911NF-12-1-0523 and NSF grant number INSPIRE-1551064. The work is supported in part by the AT\&T Foundry Innovation Centers through INQNET, a program for accelerating quantum technologies. We wish to thank the Advanced Scientific Computing Research program of the DOE for the opportunity to first present and discuss this work at the ASCR workshop on Quantum Computing for Science (2015). We acknowledge the funding agencies and all the scientists and staff at CERN and internationally whose hard work resulted in the momentous H(125) discovery in 2012.

Contact: Joshua Job, jjob@usc.edu
Department of Physics, University of Southern California
University of Southern California